
TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: M2612 – Elektrotechnika a informatika

Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika

**Využití umělých neuronových sítí pro řízení
mechanických systémů s neznámou vnitřní
strukturou**

**Using artificial neural networks for control
of mechanical systems with unknown inner
structure**

Diplomová práce

Autor:

Radek Srb

Vedoucí práce:

Doc. Ing. Jan Cvejn, Ph.D.

V Liberci 18. 5. 2006

Zadání

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé diplomové práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Rád bych poděkoval Doc. Ing. Janu Cvejnovi, Ph.D. za podporu, vedení a cenné rady při tvorbě diplomové práce.

Anotace

Cílem diplomové práce je ověřit možnost použití umělých neuronových sítí pro řízení mechanických systémů s tuhými členy, bez znalosti jejich vnitřní struktury. Řízení by mělo vycházet pouze ze závislosti dynamiky koncového bodu mechanismu na řízení. Pro získání této závislosti, z pohybu systému, má být použita neuronová síť.

Byla implementována vícevrstvá perceptronová síť a algoritmy na její trénování. Byly navrženy postupy pro získání trénovacích dat a následné jejich využití pro řízení mechanismů. Funkce byla ověřena na matematických modelech několika rovinných robotických systémů. Vše je demonstrováno na vytvořené aplikaci, která umožňuje simulaci a řízení těchto modelů v reálném čase.

Abstract

The aim of diploma thesis is to verify possibility usage of artificial neural networks for control mechanical systems with solid elements, where their inner structure is not known. Control should go out only from dependency dynamics of end point on control. For getting this dependency, from the system movement, the neural network will be used.

The multilayer perceptron neural network and algorithms for its training were implemented. The technique were design for getting training data and their following utilization for control mechanisms. The function was verified on mathematical models of some planar robotics systems. Everything is demonstrated at the application, which is able to simulating and controlling these models in real time.

Obsah

Seznam ilustrací.....	7
Úvod.....	8
1 Neuronové sítě.....	9
1.1 Neuronová síť.....	9
1.2 Vícevrstvá perceptronová síť.....	9
1.2.1 Struktura sítě.....	10
1.2.2 Vybavování ve vícevrstvé perceptronové síti.....	12
1.2.3 Učení ve vícevrstvé perceptronové síti.....	12
1.2.4 Metoda zpětného šíření (backpropagation).....	13
1.2.5 Metody pro optimalizaci chyby sítě.....	14
2 Mechanické systémy a jejich modely.....	17
2.1 Mechanické systémy.....	17
2.2 Modely.....	18
2.2.1 Obecný model.....	19
2.2.2 Model ortogonálního robotu.....	20
2.2.3 Model angulárního robotu.....	21
2.2.4 Model obsahující posun následovaný rotací.....	24
2.2.5 Model obsahující rotaci následovanou posunem.....	27
3 Princip řízení.....	29
3.1 Klasické řízení robotů.....	29
3.2 Aproximace dynamiky systému neuronovou sítí.....	31
3.2.1 Využití aproximace dynamiky pro řízení.....	31
3.2.2 Výběr vzorů pro trénování sítě.....	33
3.2.3 Volba konfigurace sítě.....	34
3.3 Zákon řízení.....	35
3.4 Trénování online.....	38
4 Praktická realizace.....	40
4.1 Aplikace NControl.....	40
4.2 Dosažené průběhy řízení.....	41
Závěr.....	45
Seznam použité literatury.....	46

Seznam ilustrací

Obr. 1.1: Formální neuron.....	9
Obr. 1.2: Vícevrstvá perceptronová síť.....	11
Obr. 1.3: Kroky metody PARTAN.....	15
Obr. 1.4: Průběhy minimalizace chyby.....	16
Obr. 2.1: Mechanický systém.....	17
Obr. 2.2: Ortogonální robot.....	20
Obr. 2.3: Angulární robot.....	21
Obr. 2.4: Model obsahující posun následovaný rotací.....	24
Obr. 2.5: Model obsahující rotaci a posun.....	27
Obr. 3.1: Řízení pohybu robotu.....	29
Obr. 3.2: Schéma řízení.....	31
Obr. 3.3: Schéma řízení s korekcí.....	33
Obr. 3.4: Průběh řešení (3.10).....	37
Obr. 3.5: Průběhy buzení, rychlosti a polohy.....	39
Obr. 4.1: Aplikace NControl.....	40
Obr. 4.2: Řízení u1.....	41
Obr. 4.3: Řízení u2.....	42
Obr. 4.4: Rychlosti v zobecněných souřadnicích.....	42
Obr. 4.5: Pozice v zobecněných souřadnicích.....	43
Obr. 4.6: Pozice v pracovním prostoru.....	43
Obr. 4.7: Trajektorie v pracovním prostoru.....	44
Tab. 1: Chyby a trénovací časy.....	35

Úvod

Při klasickém řízení robotických systému je nejprve nutné sestavit jejich matematické modely, z nich odvodit rovnice kinematiky a dynamiky. Dále pak vyřešit inverzní úlohu kinematiky, pomocí níž jsou kinematické veličiny v koncovém bodu robotu přepočítány na kinematické veličiny jednotlivých členů. Tímto postupem je možné dosáhnout velmi kvalitní a přesné řízení, avšak odvození rovnic pro složitější struktury je obtížné a získané vztahy popisují přesně pouze matematické modely, které jsou jen určitou aproximací skutečných robotů.

Naopak člověk využívá při pohybu, např. ruky, centrální nervovou soustavu, která se v průběhu života naučila závislost mezi dynamikou ruky a vzruchy, kterými ovládá svaly. Člověk, při pohybu rukou, využívá především vizuální kontrolu polohy konce ruky, případně hmat. Bez použití těchto smyslů není schopen přesného pohybu, z čehož lze usuzovat, že v jednotlivých kloubech nejsou přesná čidla polohy.

Cílem této práce je ověřit možnost užití umělých neuronových sítí, které jsou inspirovány funkcí centrální nervové soustavy živočichů, pro řízení mechanických systémů s tuhými členy, bez znalosti jejich vnitřní struktury. Řízení by mělo vycházet pouze ze závislosti dynamiky koncového bodu mechanismu na řízení. Řízení by mělo být ověřeno na simulačních modelech těchto mechanických systémů.

První kapitola se zabývá popisem neuronových sítí. Důraz je kladen především na vícevrstvou perceptronovou síť.

Ve druhé kapitole jsou popsány mechanické systémy s tuhými členy a modely pro simulaci těchto systémů.

Stěžejní kapitolou práce je kapitola třetí. Nejprve je objasněno klasické řízení robotů. Dále jsou popsány vytvořené postupy, umožňující řízení mechanických systémů, pomocí neuronové sítě.

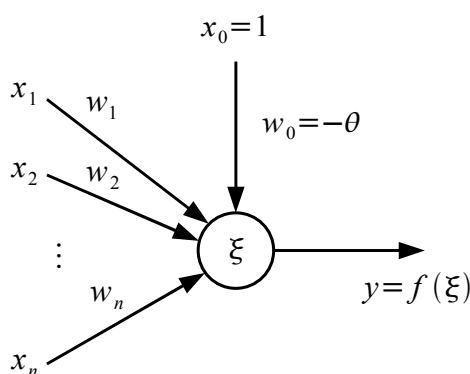
Poslední kapitolu tvoří popis aplikace NControl, která demonstruje vytvořené postupy. Na závěr jsou uvedeny dosažené průběhy řízení.

1 Neuronové sítě

Tato kapitola vychází především z publikací [1] a [2].

1.1 Neuronová síť

Neuronová síť je systém sestávající se z výpočetních jednotek neuronů, viz Obr.1.1, kde x_1, \dots, x_n jsou vstupy, w_1, \dots, w_n jsou váhy spojů, $x_0=1$ formální vstup, θ práh, $w_0=-\theta$ bias, ξ vnitřní potenciál a $y=f(\xi)$ výstup neuronu získaný aplikací aktivační funkce na potenciál, které jsou mezi sebou vzájemně propojeny spoji ohodnocenými vahami. Schopností adaptovat tyto váhy, učit se, na základě trénovacích vzorů umožňují realizovat kvalitativně novou funkci implicitně obsaženou v trénovacích datech. Důležitou vlastností neuronových sítí je kromě schopnosti učit se, tedy nacházet závislosti v trénovacích datech a ty reprezentovat pomocí vah, také schopnost zevšeobecňování získaných poznatků, tedy schopnost správně reagovat i na neznámé vstupy, na které nebyla neuronová síť naučena. Umělé neuronové sítě jsou inspirovány poznatky z oblasti neurofyzologie, tedy funkcí nervových buněk a mozku živých organismů.



Obr.1.1: Formální neuron

1.2 Vícevrstvá perceptronová síť

Vícevrstvá perceptronová síť patří mezi nejznámější a nejpoužívanější neuronové sítě. Je to síť s učitelem, tj. trénovací vzory musí kromě vstupních hodnot obsahovat i hodnoty příslušných odpovídajících výstupů.

1.2.1 Struktura sítě

Vícevrstvá perceptronová síť obsahuje z hlediska struktury neurony – perceptrony, které jsou mezi sebou propojeny tak, že vytváření vrstevnatou síť.

Perceptron je speciálním případem formálního neuronu představující obecný výpočetní prvek všech neuronových sítí, viz Obr.1.1, u něhož je vnitřní potenciál počítán jako vážený součet vstupů

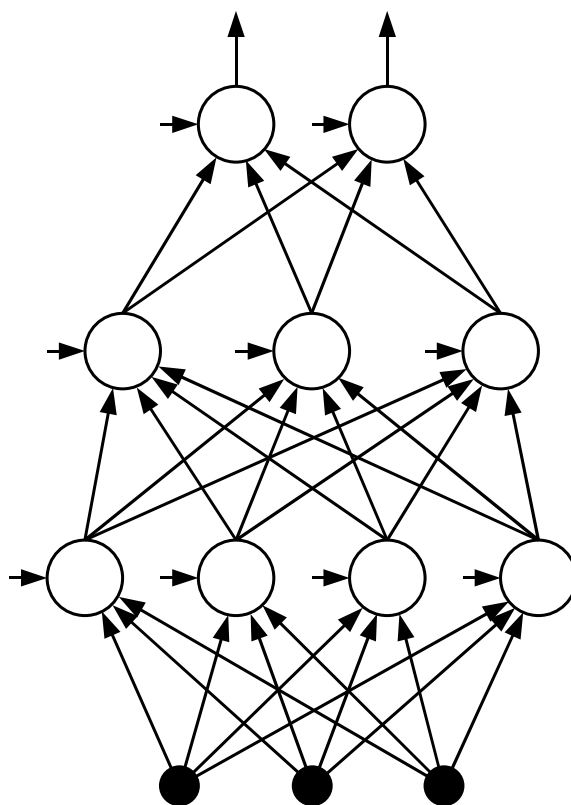
$$\xi = \sum_{i=1}^n w_i x_i - \theta = \sum_{i=0}^n w_i x_i$$

a aktivační funkcí je sigmoida

$$f(\xi) = \frac{1}{1 + e^{-\lambda \xi}} \quad (1.1)$$

kde λ je parametr strmosti.

Vícevrstvá perceptronová síť se skládá z několika vrstev perceptronů, viz Obr.1.2. Perceptrony jednotlivých sousedních vrstev jsou mezi sebou vzájemně propojeny tak, že tvoří úplný bipartitní graf, tj. výstup jednoho neuronu vrstvy je distribuován do výstupů všech perceptronů následující vrstvy. Jinými slovy, každý perceptron následující vrstvy má za své vstupy výstupy perceptronů z předcházející vrstvy. Stejným způsobem jsou distribuovány vstupy sítě k perceptronům první vrstvy. Poslední vrstva perceptronů se nazývá výstupní vrstva, ostatní vrstvy neuronů se nazývají skryté vrstvy. Podobně perceptrony těchto vrstev se nazývají výstupní, popř. skryté. Počty vrstev i počty neuronů v jednotlivých vrstvách jsou parametrem sítě a závisejí na konkrétní povaze úlohy. V praxi se odhadují na základě heuristik. Například pro perceptronovou síť se dvěma skrytými vrstvami a výstupní vrstvou se v první skryté vrstvě volí o něco více perceptronů, než je vstupů, a v druhé vrstvě se volí počet perceptronů jako aritmetický průměr počtu perceptronů v první skryté vrstvě a počtu výstupů. Pokud je počet perceptronů malý, síť nedokáže postihnou všechny závislosti v trénovacích datech. Pokud je naopak počet perceptronů velký, zvyšuje se doba učení a navíc vlivem nadměrného počtu trénovacích dat má síť špatnou schopnost generalizace způsobenou tzv. přeučením.



Obr.1.2: Vícevrstvá perceptronová síť

Pokud známe charakter úlohy a tedy strukturu trénovacích dat, můžeme vyjít při návrhu struktury sítě z vlastností, které uspořádání perceptronů umožňuje. Každý perceptron počítá vážený součet vstupů, tedy lineární kombinaci vstupů, což odpovídá rovnici nadroviny. Počet vstupů neuronu určuje stupeň této nadroviny. Pro dva vstupy dostáváme rovnici přímky, pro tři vstupy rovnici roviny atd. Proto počet neuronů první vrstvy určuje počet nadrovin, kterými lze oddělit jednotlivé třídy trénovacích dat. Další vrstva počítá kombinaci výstupů předchozí vrstvy a je tedy možné zkombinovat nadroviny předchozí vrstvy do otevřených nebo uzavřených konvexních útvarů. Další vrstva počítá opět kombinaci výstupů předchozí vrstvy a je tedy možné zkombinovat konvexní útvary předchozí vrstvy a vytvořit tak složitější nekonvexní útvary, které se mohou například protínat, překrývat, nebo mohou být úplně oddělené. Takto lze pokračovat až k výstupní vrstvě.

Vícevrstvá perceptronová síť pracuje s reálnými hodnotami, tj. všechny vstupy, výstupy, stavy, váhy i potenciály jsou obecně reálnými čísly. K tomu, aby síť správně fungovala, musí být vhodně nastaveny váhy sítě. Způsob jak nastavit tyto váhy je

předmětem učení sítě. Vzhledem k tomu, že učení sítě využívá vybavovací fázi, bude popsáno nejprve vybavování a poté teprve učení.

1.2.2 Vybavování ve vícevrstvé perceptronové síti

Ve fázi vybavování se přiloží na vstupy sítě vzor a postupně se dopředně šíří přes váhy neuronů v jednotlivých vrstvách a využitím aktivačních funkcí až k výstupům sítě. Označme písmenem n počet vstupů a písmenem m počet výstupů sítě. Pro jednoduchost zápisu označme symbolem w_{ij} váhy mezi i -tým a j -tým neuronem, kde i a j jsou indexy probíhající přes neurony dvou sousedních vrstev. Postupně se počítají hodnoty potenciálů od první k poslední vrstvě podle vztahu

$$\xi_j = \sum_i w_{ij} y_i, \quad (1.2)$$

kde index i probíhá přes perceptrony předcházející vrstvy (pro první vrstvu probíhá index i přes vstupy sítě) a na jejich základě se vypočítávají hodnoty výstupů těchto perceptronů, abychom získali hodnoty vstupů pro další vrstvu a mohli tak uskutečnit výpočet v další vrstvě podle vztahu

$$y_j = f(\xi_j), \quad (1.3)$$

kde aktivační funkce f je sigmoida (1.1). (Obecně lze za aktivační funkci f zvolit libovolnou diferencovatelnou funkci.) Parametr strmosti λ určuje míru změny nárůstu sigmoidy v okolí počátku a většinou bývá stejný a pevný v rámci celé sítě. Jeho hodnota se obvykle pohybuje blízko jedné. Hodnoty výstupů z poslední vrstvy jsou přímo výstupy sítě. Tyto hodnoty jsou obecně reálná čísla, a proto, pokud mají tyto hodnoty reprezentovat třídy při klasifikační úloze ve fázi vybavování (nikoliv učení), je třeba je vhodně kvantifikovat, např. prahováním pomocí aktivační funkce charakteru ostré nelinearity ve výstupních neuronech.

1.2.3 Učení ve vícevrstvé perceptronové síti

Snahou učení je dosáhnout takového nastavení vah, aby odchylka (chyba) mezi aktuálními a požadovanými výstupy sítě byla minimální vzhledem k odpovídajícím předloženým trénovacím vzorům. Z toho důvodu definujeme chybu sítě předpisem

$$E = \sum_k E_k, \quad (1.4)$$

kde index k probíhá přes všechny trénovací vzory a E_k je chyba odpovídající k -tému trénovacímu vzoru definovaná vztahem

$$E_k = \frac{1}{2} \sum_j (y_j - d_{kj})^2, \quad (1.5)$$

v němž index j probíhá přes neurony výstupní vrstvy a d_{kj} je j -tý element požadovaného výstupu k -tého trénovacího vzoru.

Pro optimalizaci chyby se používají gradientní metody, které budou popsány dále. Na začátku učení se všechny váhy včetně biasů nastaví na malé náhodné hodnoty se střední hodnotou v okolí nuly, např. z heuristicky doporučeného intervalu

$$\left\langle -\frac{2}{s}, \frac{2}{s} \right\rangle, \text{ kde } s \text{ je počet vstupů do neuronu, pro který váhy nastavujeme. Poté se}$$

předkládají jednotlivé trénovací vzory, pro každý se spočítá pomocí vybavovací fáze chyba podle vztahu (1.5) a tato chyba se akumuluje. Po vyčerpání všech trénovacích vzorů dostaneme výslednou chybu ze vztahu (1.4) a na jejím základě se upraví hodnoty vah sítě.

1.2.4 Metoda zpětného šíření (backpropagation)

Učící pravidlo pro vícevrstvou perceptronovou síť se nazývá metoda zpětného šíření (backpropagation). Odvození tohoto pravidla je uvedeno v následujícím textu.

Vzhledem k lineárnosti operátoru derivace a vztahu (1.4) platí, že

$$\frac{\partial E}{\partial w_{ij}} = \sum_k \frac{\partial E_k}{\partial w_{ij}}. \quad (1.6)$$

Využijeme pravidla o derivaci složené funkce a získáme vztah

$$\frac{\partial E_k}{\partial w_{ij}} = \frac{\partial E_k}{\partial y_j} \frac{\partial y_j}{\partial \xi_j} \frac{\partial \xi_j}{\partial w_{ij}}. \quad (1.7)$$

Parciální derivaci $\frac{\partial \xi_j}{\partial w_{ij}}$ obdržíme derivací vztahu pro vnitřní potenciál (1.2)

$$\frac{\partial \xi_j}{\partial w_{ij}} = y_i.$$

Parciální derivaci $\frac{\partial y_j}{\partial \xi_j}$ dostaneme derivací aktivační funkce (1.1) a využitím její vlastní funkční hodnoty

$$\frac{\partial y_j}{\partial \xi_j} = \frac{\lambda e^{-\lambda \xi_j}}{(1 + e^{-\lambda \xi_j})^2} = \left| 1 - y_j = \frac{e^{-\lambda \xi_j}}{1 + e^{-\lambda \xi_j}} \right| = \lambda \frac{e^{-\lambda \xi_j}}{1 + e^{-\lambda \xi_j}} \frac{1}{1 + e^{-\lambda \xi_j}} \lambda y_j (1 - y_j) \quad .$$

Hodnotu parciální derivace $\frac{\partial E_k}{\partial y_j}$ získáme metodou zpětného šíření, tj. postupným procházením sítě od výstupní vrstvy ke vstupní vrstvě a výpočtem těchto derivací na základě dostupné informace z vrstvy zpracované v předchozím kroku. To, že postupujeme odzadu je dáno tím, že primární informaci o chybě máme pouze ve výstupní vrstvě a do ostatních vrstev ji musíme teprve postupně přenést. Parciální derivace chyby ve výstupní vrstvě je dána jako rozdíl mezi aktuálním a požadovaným výstupem

$$\frac{\partial E_k}{\partial y_j} = y_j - d_{kj} \quad ,$$

kde index j prochází přes neurony výstupní vrstvy. Parciální derivace chyby pro skryté vrstvy počítáme podle vztahu

$$\frac{\partial E_k}{\partial y_j} = \sum_r \frac{\partial E_k}{\partial y_r} \frac{\partial y_r}{\partial \xi_r} \frac{\partial \xi_r}{\partial y_j} = \sum_r \frac{\partial E_k}{\partial y_r} \lambda y_r (1 - y_r) w_{rj} \quad ,$$

kde index r prochází přes všechny neurony, do nichž vede výstup z neuronu j a

$\frac{\partial E_k}{\partial y_r}$ jsou již známé hodnoty z předchozího výpočtu.

Dosazením příslušných parciálních derivací do vztahů (1.7) a (1.6) získáme výslednou hodnotu gradientu chybové funkce.

1.2.5 Metody pro optimalizaci chyby sítě

Pro optimalizaci chyby sítě bylo vyzkoušeno několik algoritmů. Nejjednodušší spočívá v úpravě vah sítě podle vztahu

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} \quad ,$$

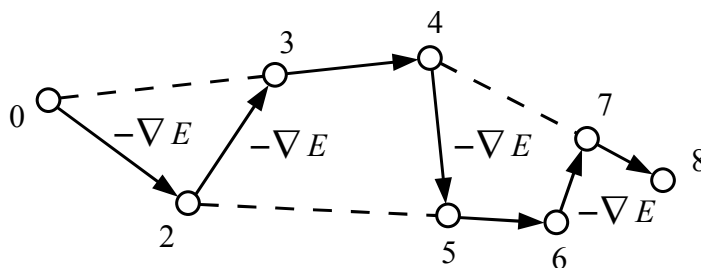
kde změny vah získáme ze vztahu

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}},$$

kde η je parametr učení, $0 \leq \eta \leq 1$. Parciální derivace chyby podle vah sítě uvedená v tomto vztahu představuje minimalizaci chyby pomocí vah gradientní metodou. Nevýhodou této metody je nutnost zvolit správně parametr η , příliš malé hodnoty způsobí pomalou konvergenci chyby sítě, v případě volby velkých hodnot chyba sítě diverguje.

Jako další byla vyzkoušena metoda největšího spádu popsaná ve skriptech [4]. Velikost η se v každém iteračním kroku k volí tak, aby bylo dosaženo ve směru záporného gradientu chyby sítě minimální hodnoty. Pro hledání minima v daném směru bylo použito algoritmu zlatého řezu, který byl rozšířen o dopředné hledání.

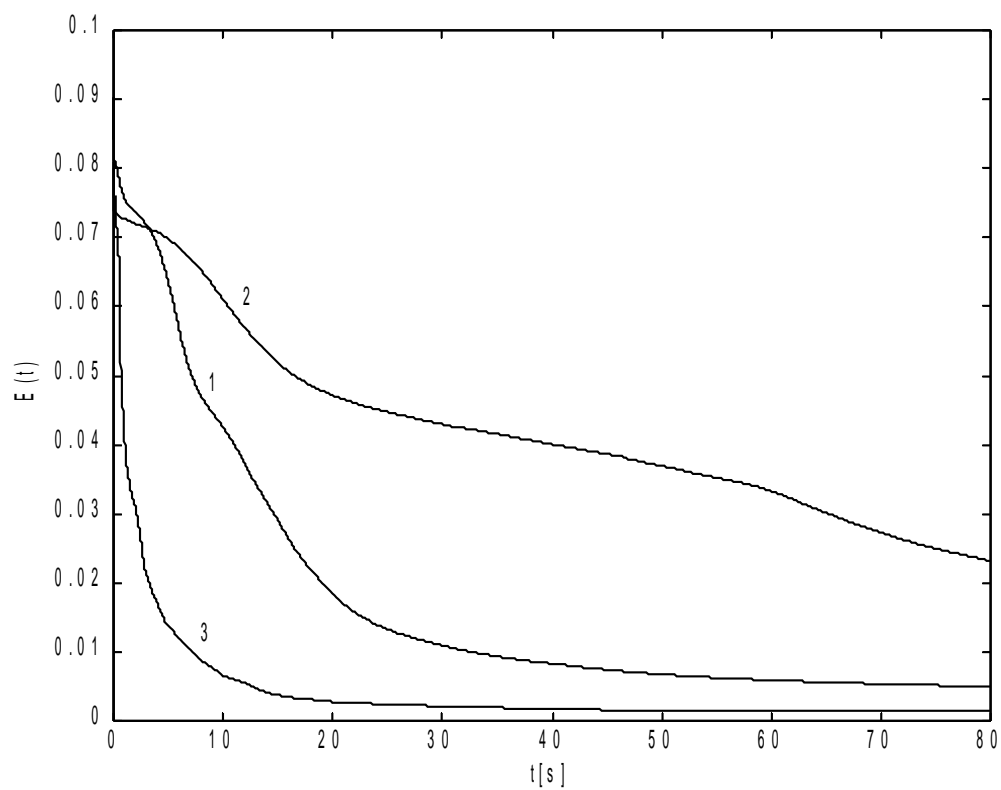
Dále byla vyzkoušena metoda PARTAN [4], znázorněna na Obr. 1.3, která je motivována kompenzací střídání kolmých směrů v metodě nejmenšího spádu.



Obr. 1.3: Kroky metody PARTAN

Při použití této metody bylo dosaženo nejrychlejší konvergence a proto byla dále používána.

Na Obr.1.4 jsou zobrazeny průběhy minimalizace chyby, pro výše popsané metody, v závislosti na čase.

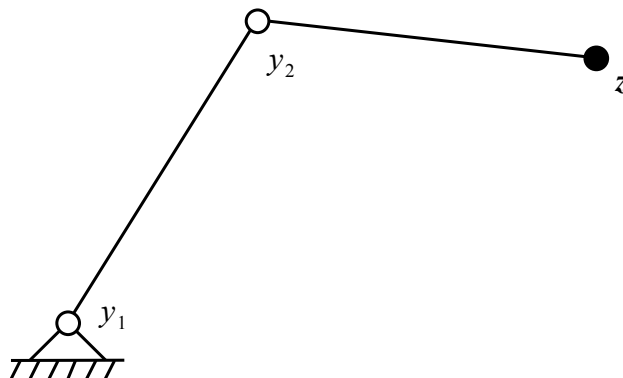


Obr.1.4: Průběhy minimalizace chyby; 1 – gradientu, 2 – největšího spádu, 3 – PARTAN

2 Mechanické systémy a jejich modely

2.1 Mechanické systémy

V této práci jsou uvažovány mechanické systémy s pevnými členy, které jsou spojeny do otevřeného řetězce, viz Obr. 2.1, příkladem takovýchto systémů jsou průmyslové roboty.



Obr. 2.1: Mechanický systém

Tyto systémy se skládají z jednotlivých členů, které jsou propojeny kinematickými vazbami, vzhledem k praktické realizovatelnosti, jsou uvažovány pouze posun a rotace. V každé této vazbě působí hnací síla popř. moment.

Pozice těchto systémů je určena vektorem zobecněných souřadnic

$$\mathbf{y} = (y_1, y_2, \dots, y_m)^T,$$

koncový bod systému je popsán vektorem

$$\mathbf{z} = (z_1, z_2, \dots, z_n)^T.$$

Transformace ze zobecněných souřadnic \mathbf{y} do souřadnic koncového bodu \mathbf{z} je určena funkcí

$$\mathbf{z} = \mathbf{f}(\mathbf{y}). \quad (2.1)$$

Dále budou uvažovány pouze systémy u kterých je funkce f prostá, tzn. každá pozice koncového bodu lze jednoznačně převést na zobecněné souřadnice. Toho lze dosáhnout tím, že zvolíme počet stupňů volnosti systému stejný, jako je počet dimenzí pracovního prostoru, ve kterém se pohybuje koncový bod. Také je nutné omezit pohyb v zobecněných souřadnicích tak, aby funkce (2.1) byla v celém pracovním prostoru prostá.

Dynamika těchto systémů je popsána rovnicí

$$\mathbf{g}(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \mathbf{F}, \quad (2.2)$$

kde \mathbf{F} je zobecněná hnací síla působící v kinematických vazbách, která je určena z buzení \mathbf{u} rovnicí

$$\mathbf{F} = \text{diag}(\mathbf{u}) \mathbf{a} + \mathbf{b}, \quad (2.3)$$

kde

$$\mathbf{a} = (a_1, a_2, \dots, a_m)^T$$

je vektor převodových poměrů mezi buzením a zobecněnými silami a

$$\mathbf{b} = (b_1, b_2, \dots, b_m)^T$$

je vektor sil nezávislých na buzení. Převodové poměry je možné volit, tak, že složky \mathbf{u} nabývají hodnot $\langle -1; 1 \rangle$. Rovnice (2.2) lze získat z Lagrangeových rovnic

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\mathbf{y}}} - \frac{\partial L}{\partial \mathbf{y}} = \mathbf{F},$$

kde

$$L = K - P$$

je rozdíl mezi kinetickou a potenciální energií.

2.2 Modely

V této kapitole bude nejprve popsán obecný simulační model uvažovaných mechanických systémů a následně pak odvození rovnic dynamiky konkrétních modelů.

Byly vytvořeny pouze modely rovinných mechanismů a to především z omezujících časových důvodů. Prostorové modely by se vytvořily dosazením rovnic jejich dynamiky do obecného modelu a dále doplněním jejich grafické reprezentace.

2.2.1 Obecný model

Pro účely simulace se rovnice dynamiky (2.2) převede do tvaru

$$\ddot{\mathbf{y}} = \mathbf{h}(\mathbf{y}, \dot{\mathbf{y}}, \mathbf{F}) \quad , \quad (2.4)$$

v této podobě je vyjádřeno zrychlení v zobecněných souřadnicích, jako funkce pozice, rychlosti a hnací síly. Nyní je možné sestavit z rovnice (2.4) stavový popis ve tvaru

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{F}) \quad ,$$

kde

$$\mathbf{x} = (y_1, \dot{y}_1, y_2, \dot{y}_2, \dots, y_m, \dot{y}_m)^T \quad .$$

Pro řešení rovnic dynamiky byla zvolena metoda Runge-Kutta řádu 4, viz [3], která je popsána následujícími vztahy:

$$\mathbf{y}_{j+1} = \mathbf{y}_j + h \left(\frac{\mathbf{k}_1}{6} + \frac{\mathbf{k}_2}{3} + \frac{\mathbf{k}_3}{3} + \frac{\mathbf{k}_4}{6} \right) \quad ,$$

kde

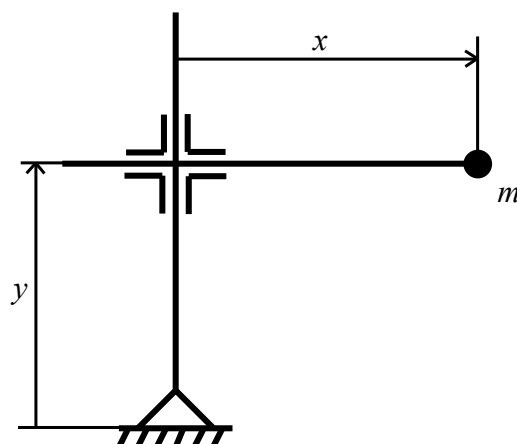
$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(\mathbf{x}, \mathbf{F}) \\ \mathbf{k}_2 &= \mathbf{f}\left(\mathbf{x} + \frac{h}{2} \mathbf{k}_1, \mathbf{F}\right) \\ \mathbf{k}_3 &= \mathbf{f}\left(\mathbf{x} + \frac{h}{2} \mathbf{k}_2, \mathbf{F}\right) \\ \mathbf{k}_4 &= \mathbf{f}(\mathbf{x} + h \mathbf{k}_3, \mathbf{F}) \end{aligned}$$

a h je velikost kroku simulace.

Dále bylo nutné nějakým vhodným způsobem ošetřit situaci, kdy se model dostane do krajní pozice. Bylo vyzkoušeno několik možností, z nichž nejjednodušší je, (v případě, že pozice v některé zobecněné souřadnici překročí stanovenou mez) tuto souřadnici nastavit zpět na mezní hodnotu. Tento způsob se ale neosvědčil, protože zůstal pohyb v této souřadnici, který způsoboval nežádoucí zrychlení v některé jiné souřadnici. Další možností je zastavit také pohyb v této souřadnici, to také nebylo vhodné, neboť v systému vzrůstala kinetická energie. Jako nejlepší řešení se jevílo, zastavit pohyb ve všech členech a tento postup byl také nakonec použit.

2.2.2 Model ortogonálního robotu

Jedná se o mechanismus, který je jednoduché řídit, protože rovnice popisující dynamiku jsou lineární a pohyb v jednotlivých osách je na sobě nezávislý. Model je znázorněn na Obr. 2.2, obsahuje pouze jednu hmotu m , která je umístěná na konci mechanismu.



Obr. 2.2: Ortogonální robot

Rovnice dynamiky je snadné získat a jsou:

$$\ddot{x} = \frac{F_x}{m}$$

$$\ddot{y} = \frac{F_y}{m} - g \quad ,$$

kde g je gravitační zrychlení.

Konkrétní hodnoty mechanických konstant byly zvoleny následovně:

$$m_1 = 1 \text{ kg}$$

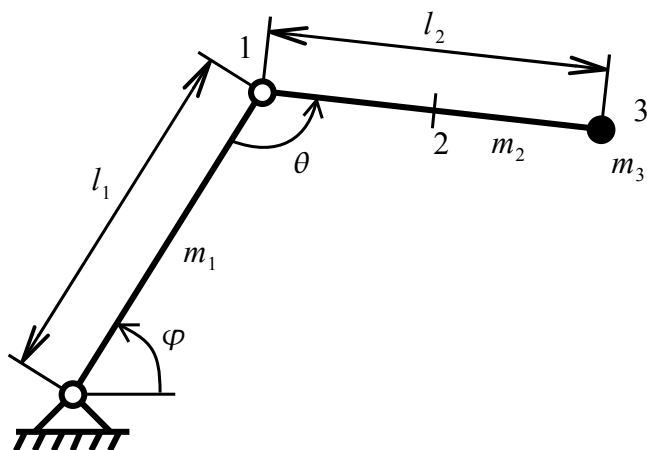
Rozsah pohybu v jednotlivých osách je:

$$x \in \langle 0,1 ; 0,6 \rangle \text{ m}$$

$$y \in \langle 0,1 ; 0,6 \rangle \text{ m}$$

2.2.3 Model angulárního robotu

Schéma angulárního robotu je zobrazeno na Obr. 2.3. Tento mechanismus je již značně nelineární a proto i jeho řízení je náročné. Jednotlivé členy, propojené rotačními vazbami, jsou uvažovány jako hmotné tyče hmotnosti m_1 a m_2 . V koncovém bodu mechanismu je umístěna hmota m_3 .



Obr. 2.3: Angulární robot

Odvození rovnic dynamiky je již o poznání složitější, než u předchozího modelu. Souřadnice bodů 1, 2 a 3 v prostoru, v závislosti na souřadnicích systému $\mathbf{y} = (\varphi, \theta)^T$, jsou:

$$X_1 = l_1 \cos(\varphi)$$

$$Y_1 = l_1 \sin(\varphi)$$

$$X_2 = l_1 \cos(\varphi) - \frac{l_2}{2} \cos(\varphi + \theta)$$

$$Y_2 = l_1 \sin(\varphi) - \frac{l_2}{2} \sin(\varphi + \theta)$$

$$X_3 = l_1 \cos(\varphi) - l_2 \cos(\varphi + \theta)$$

$$Y_3 = l_1 \sin(\varphi) - l_2 \sin(\varphi + \theta)$$

Složky vektoru rychlosti bodů 2 a 3 jsou:

$$\dot{X}_2 = -l_1 \dot{\varphi} \sin(\varphi) + \frac{l_2}{2} (\dot{\varphi} + \dot{\theta}) \sin(\varphi + \theta)$$

$$\dot{Y}_2 = l_1 \dot{\varphi} \cos(\varphi) - \frac{l_2}{2} (\dot{\varphi} + \dot{\theta}) \cos(\varphi + \theta)$$

$$\dot{X}_3 = -l_1 \dot{\varphi} \sin(\varphi) + l_2 (\dot{\varphi} + \dot{\theta}) \sin(\varphi + \theta)$$

$$\dot{Y}_3 = l_1 \dot{\varphi} \cos(\varphi) - l_2 (\dot{\varphi} + \dot{\theta}) \cos(\varphi + \theta)$$

Vyjádříme-li $\dot{X}_2^2 + \dot{Y}_2^2$ a $\dot{X}_3^2 + \dot{Y}_3^2$:

$$\dot{X}_2^2 + \dot{Y}_2^2 = l_1^2 \dot{\varphi}^2 + \frac{l_2^2}{2} (\dot{\varphi} + \dot{\theta})^2 - l_1 l_2 \dot{\varphi} (\dot{\varphi} + \dot{\theta}) \cos(\theta)$$

$$\dot{X}_3^2 + \dot{Y}_3^2 = l_1^2 \dot{\varphi}^2 + l_2^2 (\dot{\varphi} + \dot{\theta})^2 - 2l_1 l_2 \dot{\varphi} (\dot{\varphi} + \dot{\theta}) \cos(\theta) ,$$

dostaneme kinetickou energii systému ve tvaru:

$$K = m_1 l_1^2 \dot{\varphi}^2 + \frac{1}{4} m_2 l_2^2 (\dot{\varphi} + \dot{\theta})^2 + \frac{1}{2} m_2 (\dot{X}_2^2 + \dot{Y}_2^2) + \frac{1}{2} m_3 (\dot{X}_3^2 + \dot{Y}_3^2) .$$

Potenciální energie je:

$$P = g \left[m_1 \frac{l_1}{2} \sin(\varphi) + m_2 Y_2 + m_3 Y_3 \right] ,$$

kde g je tíhové zrychlení. Lagrangeovy rovnice mají tvar:

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\mathbf{y}}} - \frac{\partial K}{\partial \mathbf{y}} + \frac{\partial P}{\partial \mathbf{y}} = \mathbf{F} , \quad (2.5)$$

kde $\mathbf{F} = (M_\varphi, M_\theta)^T$ je vektor zobecněných sil. Vyjádříme-li parciální derivace K podle $\mathbf{y}, \dot{\mathbf{y}}$:

$$\begin{aligned} \frac{\partial K}{\partial \dot{\varphi}} &= l_1^2 (2m_1 + m_2 + m_3) \dot{\varphi} + l_2^2 \left(\frac{3}{4} m_2 + m_3 \right) (\dot{\varphi} + \dot{\theta}) - \\ &\quad - l_1 l_2 (m_2 + 2m_3) \dot{\varphi} \cos(\theta) - l_1 l_2 \left(\frac{1}{2} m_2 + m_3 \right) \dot{\theta} \cos(\theta) \\ \frac{\partial K}{\partial \dot{\theta}} &= l_2^2 \left(\frac{3}{4} m_2 + m_3 \right) (\dot{\varphi} + \dot{\theta}) - l_1 l_2 \left(\frac{1}{2} m_2 + m_3 \right) \dot{\varphi} \cos(\theta) \\ \frac{\partial K}{\partial \varphi} &= 0 \\ \frac{\partial K}{\partial \theta} &= l_1 l_2 \left(\frac{1}{2} m_2 + m_3 \right) (\dot{\varphi} + \dot{\theta})^2 \sin(\theta) \end{aligned}$$

Parciální derivace P jsou:

$$\frac{\partial P}{\partial \varphi} = g \left[l_1 \left(\frac{1}{2} m_1 + m_2 + m_3 \right) \cos(\varphi) - l_2 \left(\frac{1}{2} m_2 + m_3 \right) \cos(\varphi + \theta) \right]$$

$$\frac{\partial P}{\partial \theta} = -g l_2 \left(\frac{1}{2} m_2 + m_3 \right) \cos(\varphi + \theta)$$

Po dosazení do (2.5) a zavedení pomocných konstant

$$a = 2 m_1 + m_2 + m_3$$

$$b = \frac{3}{4} m_2 + m_3$$

$$c = \frac{1}{2} m_2 + m_3$$

$$d = \frac{1}{2} m_1 + m_2 + m_3$$

a funkcí

$$A = a l_1^2 + b l_2^2 - 2 c l_1 l_2 \cos(\theta)$$

$$B = c l_1 l_2 \cos(\theta) - b l_2^2$$

$$C = -2 c l_1 l_2 \dot{\varphi} \dot{\theta} \sin(\theta) - c l_1 l_2 \dot{\theta}^2 \sin(\theta) - d g l_1 \cos(\varphi) + c g l_2 \cos(\varphi + \theta) + M_\varphi$$

$$D = \frac{c}{b} \frac{l_1}{l_2} \cos(\theta) - 1$$

$$E = \frac{c}{b} \frac{l_1}{l_2} \dot{\varphi}^2 \sin(\theta) + \frac{c}{b} g \frac{1}{l_2^2} \cos(\varphi + \theta) + \frac{M_\theta}{b l_2^2}$$

získáme pohybové rovnice systému:

$$\ddot{\varphi} = \frac{B \ddot{\theta} + C}{A}$$

$$\ddot{\theta} = D \ddot{\varphi} + E$$

Po vyjádření $\ddot{\varphi}$ a $\ddot{\theta}$ dostaneme tvar:

$$\ddot{\varphi} = \frac{B E + C}{A - B D}$$

$$\ddot{\theta} = \frac{C D + A E}{A - B D}$$

Konkrétní hodnoty mechanických konstant byly zvoleny následovně:

$$l_1 = l_2 = 0,5 \text{ m}$$

$$m_1 = m_2 = 0,5 \text{ kg}$$

$$m_3 = 1 \text{ kg}$$

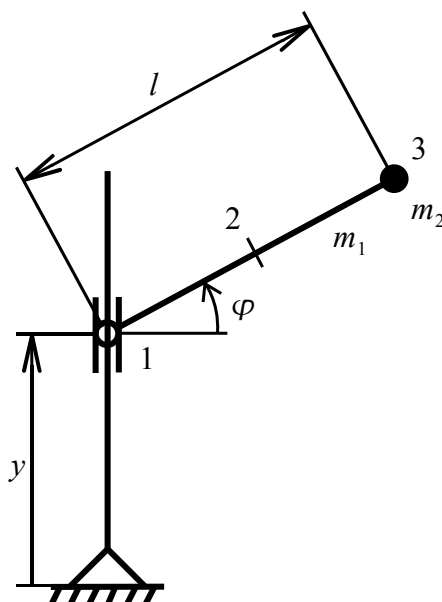
Rozsah pohybu v jednotlivých osách je:

$$\varphi \in \langle 0, 1\pi; 0,4\pi \rangle \text{ rad}$$

$$\theta \in \langle 0, 1\pi; 0,9\pi \rangle \text{ rad}$$

2.2.4 Model obsahující posun následovaný rotací

Model (Obr. 2.4) se sestává ze svislé tyče, na které je posuvně uložen člen, jehož hmotnost není uvažována a k tomuto členu je pomocí rotační vazby uchycena hmotná tyč hmotnosti m_1 a délky l . V koncovém bodě mechanismu je umístěna hmota m_2 .



Obr. 2.4: Model obsahující posun následovaný rotací

Souřadnice bodů 2 a 3 v prostoru, v závislosti na souřadnicích systému $\mathbf{y} = (y, \varphi)^T$, jsou:

$$X_2 = \frac{l}{2} \cos(\varphi)$$

$$Y_2 = y + \frac{l}{2} \sin(\varphi)$$

$$X_3 = l \cos(\varphi)$$

$$Y_3 = y + l \sin(\varphi)$$

Složky vektoru rychlosti bodů 2 a 3 jsou:

$$\begin{aligned}\dot{X}_2 &= -\frac{l}{2} \dot{\varphi} \sin(\varphi) \\ \dot{Y}_2 &= \frac{l}{2} \dot{\varphi} \cos(\varphi) + \dot{y} \\ \dot{X}_3 &= -l \dot{\varphi} \sin(\varphi) \\ \dot{Y}_3 &= l \dot{\varphi} \cos(\varphi) + \dot{y}\end{aligned}$$

Vyjádříme-li $\dot{X}_2^2 + \dot{Y}_2^2$ a $\dot{X}_3^2 + \dot{Y}_3^2$:

$$\begin{aligned}\dot{X}_2^2 + \dot{Y}_2^2 &= \frac{1}{4} l^2 \dot{\varphi}^2 + l \dot{y} \dot{\varphi} \cos(\varphi) + \dot{y}^2 \\ \dot{X}_3^2 + \dot{Y}_3^2 &= l_1^2 \dot{\varphi}^2 + l_2^2 (\dot{\varphi} + \dot{\theta})^2 - 2l_1 l_2 \dot{\varphi} (\dot{\varphi} + \dot{\theta}) \cos(\theta) \quad ,\end{aligned}$$

dostaneme kinetickou energii systému ve tvaru:

$$K = \frac{1}{2} m_1 (\dot{X}_2^2 + \dot{Y}_2^2) + \frac{1}{4} m_1 l^2 \dot{\varphi}^2 + \frac{1}{2} m_2 (\dot{X}_3^2 + \dot{Y}_3^2) \quad .$$

Potenciální energie je:

$$P = g \left[(m_1 + m_2) y + \left(\frac{1}{2} m_1 + m_2 \right) l \sin(\varphi) \right] \quad ,$$

kde g je tíhové zrychlení. Lagrangeovy rovnice mají tvar:

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\mathbf{y}}} - \frac{\partial K}{\partial \mathbf{y}} + \frac{\partial P}{\partial \mathbf{y}} = \mathbf{F} \quad , \quad (2.6)$$

kde $\mathbf{F} = (F_y, M_\varphi)^T$ je vektor zobecněných sil. Vyjádříme-li parciální derivace K podle $\mathbf{y}, \dot{\mathbf{y}}$:

$$\begin{aligned}\frac{\partial K}{\partial \dot{y}} &= \frac{1}{2} m_1 [l \dot{\varphi} \cos(\varphi) + 2 \dot{y}] + \frac{1}{2} m_2 [2 l \dot{\varphi} \cos(\varphi) + 2 \dot{y}] \\ \frac{\partial K}{\partial \dot{\varphi}} &= \frac{1}{2} m_1 \left[\frac{l}{2} l^2 \dot{\varphi} + l \dot{y} \cos(\varphi) \right] + \frac{1}{2} m_1 l^2 \dot{\varphi} + \frac{1}{2} m_2 [2 l^2 \dot{\varphi} + 2 l \dot{y} \cos(\varphi)] \\ \frac{\partial K}{\partial y} &= 0 \\ \frac{\partial K}{\partial \varphi} &= -\frac{1}{2} m_1 l \dot{y} \dot{\varphi} \sin(\varphi) - m_2 l \dot{y} \dot{\varphi} \sin(\varphi)\end{aligned}$$

Parciální derivace P jsou:

$$\frac{\partial P}{\partial y} = g(m_1 + m_2)$$

$$\frac{\partial P}{\partial \varphi} = g\left(\frac{1}{2}m_1 + m_2\right)l \cos(\varphi)$$

Po dosazení do (2.6) a zavedení pomocných konstant

$$a = m_1 + 2m_2$$

$$b = 2m_1 + m_2$$

$$c = 3m_1 + 4m_2$$

$$d = m_1 + m_2$$

získáme pohybové rovnice systému:

$$\ddot{y} = -\frac{a}{b}l\ddot{\varphi}\cos(\varphi) + \frac{a}{b}l\dot{\varphi}^2\sin(\varphi) - 2\frac{d}{b}g + \frac{2}{b}F_y$$

$$\ddot{\varphi} = -2\frac{a}{c}\frac{1}{l}\ddot{y}\cos(\varphi) - 2\frac{a}{c}g\frac{1}{l}\cos(\varphi) + \frac{4}{c}\frac{1}{l^2}M_\varphi$$

Po vyjádření \ddot{y} a $\ddot{\varphi}$ dostaneme tvar:

$$\ddot{y} = \frac{2a^2g\cos^2(\varphi) - 4a\frac{1}{l}M_\varphi\cos(\varphi) + acl\dot{\varphi}^2\sin(\varphi) - 2cdg + 2cF_y}{bc - 2a^2\cos^2(\varphi)}$$

$$\ddot{\varphi} = \frac{\cos(\varphi)\left[-2a^2\dot{\varphi}^2\sin(\varphi) + 4adg\frac{1}{l} - \frac{4a}{bc}\frac{1}{l}F_y - abg\frac{2}{l}\right] + b\frac{4}{l^2}M_\varphi}{bc - 2a^2\cos^2(\varphi)}$$

Konkrétní hodnoty mechanických konstant byly zvoleny následovně:

$$l = 0,5 \text{ m}$$

$$m_1 = 0,5 \text{ kg}$$

$$m_2 = 1 \text{ kg}$$

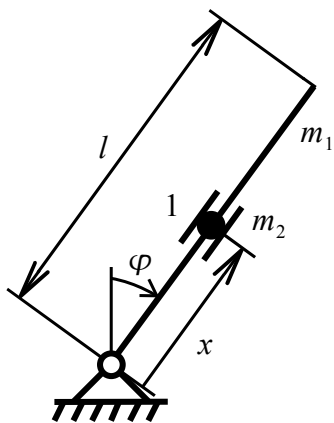
Rozsah pohybu v jednotlivých osách je:

$$y \in \langle 0, 1; 0, 6 \rangle \text{ m}$$

$$\varphi \in \langle 0, 2\pi; 0, 8\pi \rangle \text{ rad}$$

2.2.5 Model obsahující rotaci následovanou posunem

Schéma modelu je znázorněno na Obr. 2.5. Na rámu je pomocí rotační vazby přichycena hmotná tyč hmotnosti m_1 a délky l , po které se posouvá hmota m_2 .



Obr. 2.5: Model obsahující rotaci a posun

Souřadnice bodu 1 v prostoru, v závislosti na souřadnicích systému $\mathbf{y} = (\varphi, x)^T$, jsou:

$$X_1 = x \sin(\varphi)$$

$$Y_1 = x \cos(\varphi)$$

Složky vektoru rychlosti bodu 1 jsou:

$$\dot{X}_1 = \dot{x} \sin(\varphi) + x \dot{\varphi} \cos(\varphi)$$

$$\dot{Y}_1 = \dot{x} \cos(\varphi) - x \dot{\varphi} \sin(\varphi)$$

Vyjádříme-li $\dot{X}_1^2 + \dot{Y}_1^2$:

$$\dot{X}_1^2 + \dot{Y}_1^2 = \dot{x}^2 + x^2 \dot{\varphi}^2,$$

dostaneme kinetickou energii systému ve tvaru:

$$K = m_1 l^2 \dot{\varphi}^2 + \frac{1}{2} m_2 (\dot{x}^2 + x^2 \dot{\varphi}^2).$$

Potenciální energie je:

$$P = g \cos(\varphi) \left(\frac{1}{2} m_1 l + m_2 x \right),$$

kde g je tíhové zrychlení. Lagrangeovy rovnice mají tvar:

$$\frac{d}{dt} \frac{\partial K}{\partial \dot{\mathbf{y}}} - \frac{\partial K}{\partial \mathbf{y}} + \frac{\partial P}{\partial \mathbf{y}} = \mathbf{F}, \quad (2.7)$$

kde $\mathbf{F} = (M_\varphi, F_x)^T$ je vektor zobecněných sil. Vyjádříme-li parciální derivace K podle $\mathbf{y}, \dot{\mathbf{y}}$:

$$\begin{aligned} \frac{\partial K}{\partial \dot{\varphi}} &= (2m_1 l^2 + m_2 x^2) \dot{\varphi} \\ \frac{\partial K}{\partial \dot{x}} &= m_2 \dot{x} \\ \frac{\partial K}{\partial \varphi} &= 0 \\ \frac{\partial K}{\partial x} &= m_2 x \dot{\varphi}^2 \end{aligned}$$

Parciální derivace P jsou:

$$\begin{aligned} \frac{\partial P}{\partial \varphi} &= -g \sin(\varphi) \left(\frac{1}{2} m_1 l + m_2 x \right) \\ \frac{\partial P}{\partial x} &= g m_2 \cos(\varphi) \end{aligned}$$

Po dosazení do (2.7) a úpravě, získáme pohybové rovnice systému:

$$\begin{aligned} \ddot{\varphi} &= \frac{-2m_2 x \dot{x} \dot{\varphi} + g \sin(\varphi) \left(\frac{1}{2} m_1 l + m_2 x \right) + M_\varphi}{2m_1 l^2 + m_2 x^2} \\ \ddot{x} &= x \dot{\varphi}^2 - g \cos(\varphi) + \frac{F_x}{m_2}. \end{aligned}$$

Konkrétní hodnoty mechanických konstant byly zvoleny následovně:

$$\begin{aligned} l &= 0,7 \text{ m} \\ m_1 &= 0,5 \text{ kg} \\ m_2 &= 1 \text{ kg} \end{aligned}$$

Rozsah pohybu v jednotlivých osách je:

$$\begin{aligned} \varphi &\in \langle -0,3\pi; 0,3\pi \rangle \text{ rad} \\ x &\in \langle 0,1; 0,6 \rangle \text{ m} \end{aligned}$$

3 Princip řízení

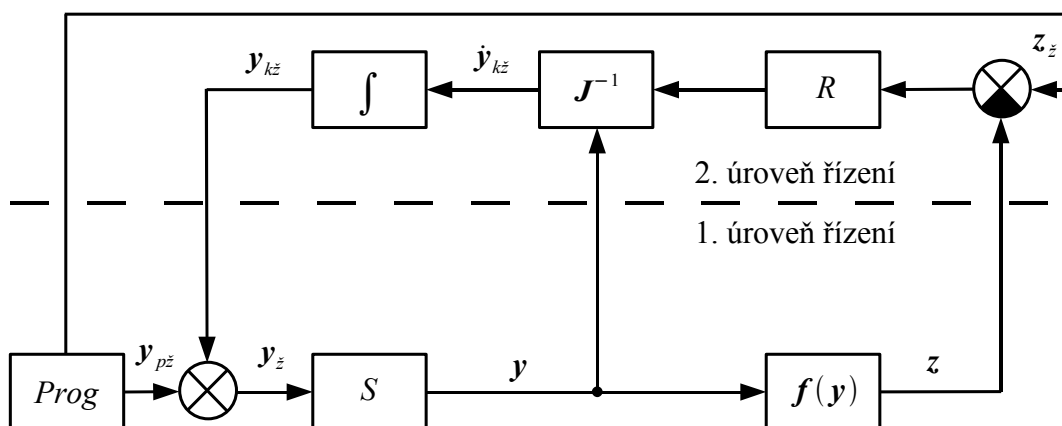
3.1 Klasické řízení robotů

Tato kapitola vychází z [5].

Každý člen robotu je pohybován mechanismem s vlastní zpětnou vazbou od polohy mechanismu – servomechanismem. Tento servomechanismus využívá při řízení efektivní moment setrvačnosti, který je vypočítán z konfigurace robotu v daném okamžiku, a to tak, aby řízení bylo co nejrychlejší a zároveň aperiodické. Kromě toho pro přesné resp. adaptivní řízení (které se přizpůsobuje situaci na pracovišti) se realizuje zpětná vazba od polohy chapadla či technologické hlavice (např. při svařování vzdálenost hlavice od místa sváru). Celkové schéma řízení je pak na Obr. 3.1, kde blok

S představuje dynamický systém robotu včetně servomechanismů a manipulovaného předmětu či technologické hlavice. Vstupem do systému je žádaná poloha výstupů jednotlivých servomechanismů vyjádřená v zobecněných souřadnicích y_z , výstupem jsou skutečné polohy servomechanismů v zobecněných souřadnicích y a poloha předmětu či technologické hlavice v souřadnicích pracoviště z , které jsou obecně nelineární funkcí souřadnic zobecněných

$$z = f(y) \text{ .}$$



Obr. 3.1: Řízení pohybu robotu

V první úrovni řízení představují servomechanismy svoji polohu podle žádaných hodnot \mathbf{y}_z . Vzhledem k dynamice systému je regulační pochod popsán soustavou nelineárních diferenciálních rovnic

$$\dot{\mathbf{y}} = \mathbf{g}(\mathbf{y}, \mathbf{y}_z) .$$

Žádané hodnoty \mathbf{y}_z jsou v první úrovni řízení vytvářeny v programovacím zařízení *Prog*. V paměti programovacího zařízení jsou uloženy souřadnice uzlových bodů trajektorie a koeficienty spline, pomocí kterých se v interpolátoru generuje trajektorie jako funkce času \mathbf{q}_{pz} .

V druhé úrovni řízení se koriguje žádaná hodnota zobecněných souřadnic podle zjištěné regulační odchylky $\mathbf{z}_z - \mathbf{z}$. Tato odchylka se zpracuje v regulátoru *R* a přepočte na korekci žádané hodnoty \mathbf{y}_{kz} . V nejjednodušším případě integračního regulátoru *I* se volí rychlost opravy úměrná okamžité odchylce

$$\dot{\mathbf{z}} = I(\mathbf{z}_z - \mathbf{z})$$

a tato se přepočte pomocí kinematických vztahů na rychlost změny korekce žádané hodnoty

$$\dot{\mathbf{y}}_z = \mathbf{J}^{-1} \dot{\mathbf{z}} ,$$

kde je

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{y})}{\partial \mathbf{y}}$$

Jacobiho matice soustavy robotu. Integrací dostáváme pak potřebnou korekci žádané hodnoty zobecněných souřadnic \mathbf{y}_{kz} .

Zmiňované kinematické vztahy se odvozují z matematických modelů uvažovaných robotů. Toto odvození je pro složitější roboty velmi obtížné a získané vztahy přesně neodpovídají skutečným robotům, protože modely jsou pouze jejich aproximací.

3.2 Aproximace dynamiky systému neuronovou sítí

3.2.1 Využití aproximace dynamiky pro řízení

V rámci řízení jsou mechanické systémy, popsané v kapitole 2, uvažovány jako diskrétní, toho lze dosáhnout nahrazením spojitého času t diskrétním časem kT

$$t = t_k = kT ,$$

kde T je perioda vzorkování. Dále je použit relativní diskrétní čas

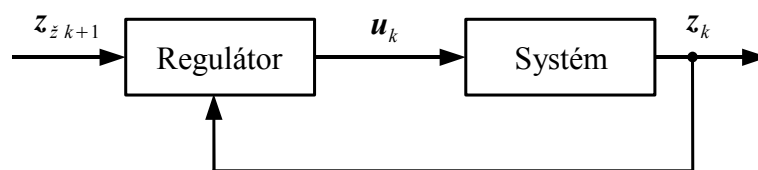
$$k = \frac{t_k}{T} .$$

Rovnice (2.4) tedy přejde, po dosazení (2.1) a (2.3), do tvaru

$$\mathbf{z}_{k+1} = \mathbf{g}_1(\mathbf{z}_k, \dot{\mathbf{z}}_k, \mathbf{u}_k) \quad (3.1)$$

$$\dot{\mathbf{z}}_{k+1} = \mathbf{g}_2(\mathbf{z}_k, \dot{\mathbf{z}}_k, \mathbf{u}_k) . \quad (3.2)$$

Pro řízení by bylo vhodné, kdyby existovala funkce, která by poskytovala aktuální hodnotu řízení, které by uvedlo systém do požadované polohy v příštím kroku. Dále by bylo vhodné, aby tato funkce vycházela pouze z hodnot polohy a řízení v předchozích krocích. Pak by bylo možné řízení provést podle schéma na Obr. 3.2, kde \mathbf{z}_k je aktuální poloha, \mathbf{u}_k aktuální řízení a \mathbf{z}_{k+1} žádaná poloha v příštím kroku. V dalším odstavci bude ukázáno, že taková funkce existuje.



Obr. 3.2: Schéma řízení

Implicitní tvar (3.1) je:

$$\Phi(\mathbf{z}_{k+1}, \mathbf{z}_k, \dot{\mathbf{z}}_k, \mathbf{u}_k) .$$

Pokud, v uvažovaném definičním oboru, je

$$\frac{\partial \Phi}{\partial \mathbf{u}_k} \neq 0 ,$$

pak existuje, viz věta o implicitní funkci [6], funkce

$$\mathbf{u}_k = \boldsymbol{\gamma}(\mathbf{z}_{k+1}, \mathbf{z}_k, \dot{\mathbf{z}}_k) \quad . \quad (3.3)$$

Po dosazení $k-1$ za k do (3.1) dostaneme tvar

$$\mathbf{z}_k = \mathbf{h}_1(\mathbf{z}_{k-1}, \dot{\mathbf{z}}_{k-1}, \mathbf{u}_{k-1}) \quad ,$$

nyiní je možné vyjádřit $\dot{\mathbf{z}}_{k-1}$ a tím získat tvar

$$\dot{\mathbf{z}}_{k-1} = \boldsymbol{\beta}(\mathbf{z}_k, \mathbf{z}_{k-1}, \mathbf{u}_{k-1}) \quad . \quad (3.4)$$

Dosadíme-li $k-1$ za k do (3.2) dostaneme tvar

$$\dot{\mathbf{z}}_k = \mathbf{h}_2(\mathbf{z}_{k-1}, \dot{\mathbf{z}}_{k-1}, \mathbf{u}_{k-1}) \quad ,$$

po dosazení (3.4) za $\dot{\mathbf{z}}_{k-1}$ a následně pak do (3.3) získáme tvar

$$\mathbf{u}_k = \tilde{\boldsymbol{\gamma}}(\mathbf{z}_{k+1}, \mathbf{z}_k, \mathbf{z}_{k-1}, \mathbf{u}_{k-1}) \quad . \quad (3.5)$$

Funkce (3.5) splňuje požadavky kladené v předchozím odstavci.

Funkce (3.5) existuje za předpokladu, že existují všechna inverzní zobrazení použitá výše, třebaže není známá. Proto byla pro řízení použita neuronová síť, která se z dat získaných při pohybu systému tuto funkci naučí aproximovat.

Na vstup sítě nejsou přiváděny přímo proměnné funkce (3.5), ale hodnoty \mathbf{z}_k , $\Delta \mathbf{z}_{k+1}$, $\Delta \mathbf{z}_k$ a \mathbf{z}_{k-1} získané následovně:

$$\Delta \mathbf{z}_{k+1} = \frac{\mathbf{z}_{k+1} - \mathbf{z}_k}{T}$$

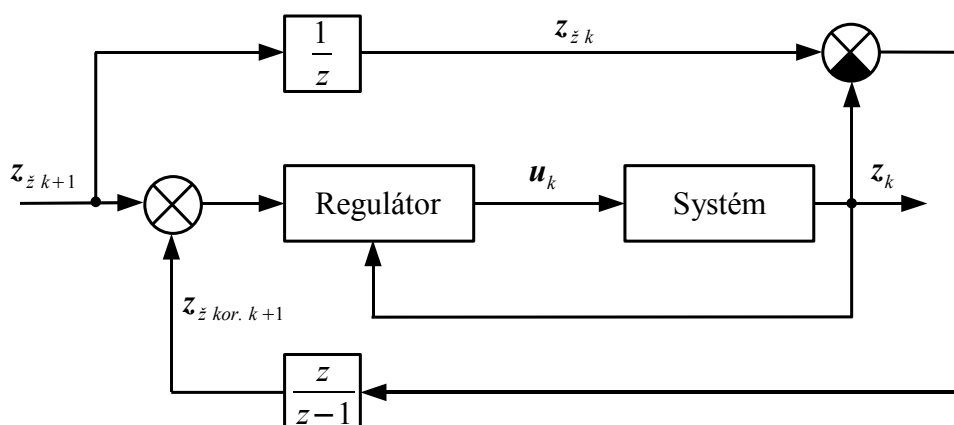
$$\Delta \mathbf{z}_k = \frac{\mathbf{z}_k - \mathbf{z}_{k-1}}{T} \quad .$$

Vstup sítě byl volen tímto způsobem, aby hodnoty vstupující do sítě byly přibližně stejných rozsahů, jinak by se trénování zkomplikovalo, protože hodnoty s větším rozsahem by měly větší „váhu“. Dále pak hodnota \mathbf{z}_k představuje polohu systému, hodnoty $\Delta \mathbf{z}_{k+1}$ a $\Delta \mathbf{z}_k$ představují předchozí a následnou změnu polohy, což lépe odpovídá stavu systému, než tři následné pozice.

Při experimentech řízení založených na schématu, viz Obr. 3.2, se ukázalo, že poloha v následném kroku \mathbf{z}_{k+1} přesně neodpovídá požadované poloze $\mathbf{z}_{\hat{k}+1}$, což způsobí trvalou regulační odchylku, která je u robotických systémů nežádoucí. Tento problém je způsoben tím, že neuronová síť aproximuje funkci (3.5) pouze s určitou

přesností. Kvalitu aproximace lze zvýšit např. úpravou struktury sítě, tím se trvalá regulační odchylka sníží, ale princip problému se neodstraní.

Možným řešením je zařazení určité korekce, která by zajistila dosažení požadované polohy. Chyba je způsobena špatnou hodnotou u_k , proto by se mohlo zdát vhodné zařadit korekci do této větve, ale vzhledem k tomu, že není znám vztah mezi chybou žádané polohy a řízením, možné to není. Možné řešení spočívá v zařazení korekce do větve požadované polohy $z_{z\ k+1}$, viz Obr. 3.3.



Obr. 3.3: Schéma řízení s korekcí

Hodnota požadované polohy v příštím kroku $z_{z\ k+1}$ je přivedena do zpožďovacího členu $\frac{1}{z}$, na jehož výstupu je požadovaná poloha v aktuálním kroku, od této hodnoty je odečtena skutečná poloha a tím získána odchylka od požadované polohy. Tato odchylka se přivede na sumační člen $\frac{z}{z-1}$, na jehož výstupu je hodnota korekce žádané polohy $z_{z\ kor. k+1}$, která se přičte k požadované poloze $z_{z\ k+1}$ a teprve tato hodnota je přiváděna na vstup regulátoru.

3.2.2 Výběr vzorů pro trénování sítě

Pro učení sítě je potřeba získat trénovací množinu, která obsahuje dvojice vstupních a požadovaných výstupních hodnot. Množina vstupních hodnot by měla pokrývat celý prostor potenciálních vstupů, přičemž navíc „hustěji“ tam, kde učená funkce vykazuje větší změny, viz [1].

Pro základní trénování sítě byl zvolen následující postup získání trénovacích dat. V pracovním prostoru systému se zvolí náhodně pozice \mathbf{z} , která je převedena na zobecněné souřadnice \mathbf{y} . Dále se náhodně zvolí rychlosti jednotlivých akčních členů $\dot{\mathbf{y}}$ z intervalu $\langle -a; a \rangle$, kde a je maximální uvažovaná rychlost členu. Poté se náhodně zvolí hodnota řízení z intervalu $\langle -1; 1 \rangle$. Následně se provede krok simulace a nastaví se nové náhodné hodnoty řízení ze stejného intervalu. Poté se opět provede další krok simulace. Tímto postupem získáme hodnoty \mathbf{z}_{k-2} , \mathbf{z}_{k-1} , \mathbf{z}_k , \mathbf{u}_{k-2} a \mathbf{u}_{k-1} , které tvoří jeden trénovací vzor.

Výše popsany postup poskytuje trénovací množinu, která rovnoměrně pokrývá celý prostor vstupů sítě. Nevýhodou této metody je její nerealizovatelnost u skutečných mechanických systémů. Je ovšem vhodná pro testování sítě, především pro zjištění je-li síť vůbec schopná se požadovanou funkci naučit a s jakou chybou.

3.2.3 Volba konfigurace sítě

Pro trénování sítě byla použita metoda PARTAN, popsaná v kapitole 1.2.5. Bylo nutné nějakým vhodným způsobem provést ukončení tohoto trénovacího algoritmu. Vzhledem k tomu, že velikost gradientu chyby sítě velmi kolísá, je algoritmus ukončen, pokud v posledních stech iteracích nedošlo k poklesu chyby sítě alespoň o jedno procento.

Dále bylo potřeba navrhnout kritérium, které by hodnotilo, jak přesně síť aproximuje učenou funkci. Pro tento účel se, postupem uvedeným v předchozí kapitole, vytvoří dostatečně rozsáhlá množina vzorů, pro kterou se určí chyba sítě. Tato chyba se, vydělením počtem vzorů, přepočítá na chybu jednoho vzoru, aby ji bylo možné porovnávat pro různé počty vzorů. Dalším hodnoceným parametrem sítě je čas, potřebný pro natrénování sítě.

Pro trénování je použita množina vzorů, která obsahuje pět krát větší počet vzorů, než je počet vah sítě. Tato hodnota byla určena experimentálně. Při použití menšího počtu vzorů se zvětšovala chyba sítě, naopak větší počet způsoboval prodlužování učení, bez snížení chyby sítě.

Byly porovnávány sítě bez, s jednou a se dvěma skrytými vrstvami. Při použití, sítě bez skryté vrstvy, probíhalo učení velmi rychle, ale pro modely nelineárních systémů byla chyba sítě značná. Síť s jednou skrytou vrstvou poskytovala, pro

nelineární modely výrazně nižší chybu, avšak čas trénování se značně prodloužil. U sítě, se dvěmi skrytými vrstvami, byla chyba nepatrně nižší, než v předchozím případě, ale doba učení neúměrně vzrostla. Zjištěné chyby a trénovací časy, pro jednotlivé sítě a modely, jsou uvedeny v Tab. 1.

Tab. 1: Chyby a trénovací časy

Model		Počet skrytých vrstev		
		0	1	2
Ortogonalní	E	0,00091	0,00088	0,00090
	t[s]	0,7	31,2	140,4
Angulární	E	0,044	0,0076	0,0060
	t[s]	0,6	128,1	466,0
Posun s rotací	E	0,023	0,0017	0,0014
	t[s]	0,6	150,3	640,4
Rotace s posunem	E	0,042	0,0076	0,006
	t[s]	0,4	102,3	590,2

3.3 Zákon řízení

V kapitole 3.2.1 je popsáno řízení systému v rámci jednoho kroku řízení. Je tedy potřeba vytvořit algoritmus, který by na základě několika předchozích hodnot polohy systému a požadované cílové pozici, generoval příslušné hodnoty polohy v konkrétních krocích řízení. Dále by bylo vhodné, aby navržený algoritmus generoval takové hodnoty, aby výsledná trajektorie koncového bodu mechanismu byla úsečkou.

Vzhledem k tomu, že rovnice popisující dynamiku uvažovaných mechanických systémů jsou druhého řádu, byla pro generování následujících pozic zvolena diferenční rovnice tvaru

$$e_{k+1} + k_1 e_k + k_0 e_{k-1} = 0 \quad , \quad (3.6)$$

kde e_k je odchylka od cílové pozice z_c a je tedy určena vztahem

$$e_k = z_c - z_k \quad .$$

Pro další úvahy je rovnice (3.6) dosazením $k+1$ za k převedena do základního tvaru

$$e_{k+2} + k_1 e_{k+1} + k_0 e_k = 0 \quad . \quad (3.7)$$

Pro správnou funkci algoritmu je nutné, aby rovnice (3.7) byla stabilní. Z-obraz rovnice (3.7) má tvar

$$z^2[E(z) - e(0) - e(1)z^{-1}] + k_1 z[E(z) - e(0)] + k_0 E_z = 0 \quad ,$$

po vyjádření $E(z)$ dostaneme

$$E(z) = \frac{e(0)z^2 + k_1 e(0)z + e(1)z}{z^2 + k_1 z + k_0} \quad . \quad (3.8)$$

Kořeny charakteristické rovnice $z^2 + k_1 z + k_0 = 0$ jsou

$$z_{1,2} = -\frac{k_1}{2} \pm \sqrt{\frac{k_1^2}{4} - k_0} \quad .$$

Kvůli požadavku na rychlé řízení, je kořen volen jako dvojnásobný

$$s = z_{1,2}$$

a tudíž je k_0 určeno vztahem

$$k_0 = \frac{k_1^2}{4} \quad .$$

Kořen s je tedy určen vztahem

$$s = -\frac{k_1}{2} \quad .$$

Dalším požadavkem je, aby byl průběh řízení aperiodický a proto je nutné s volit z intervalu $(0; 1)$, tím pádem k_1 z intervalu $(-2; 0)$. Čím více se budou volené hodnoty k_1 blížit k nule, tím bude regulační pochod rychlejší.

Při praktických experimentech se ukázalo, že při volbě konstanty k_1 blízko nuly, dochází ke kmitání systému kolem cílové pozice. Konstanta k_1 byla tedy experimentálně volena tak, aby regulace byla rychlá a zároveň systém nekmital.

Nyní bude ukázáno, že trajektorie generovaná diferenční rovnicí (3.6) je úsečkou. To je možné, pouze pokud je systém na začátku trajektorie v klidu nebo se pohybuje směrem k cílové pozici a to proto, že jde o mechanický systém, u kterého není možné měnit nespojitě rychlost. Bude tedy uvažováno, že počáteční podmínky jsou stejné

$$e(0) = e(1) \quad .$$

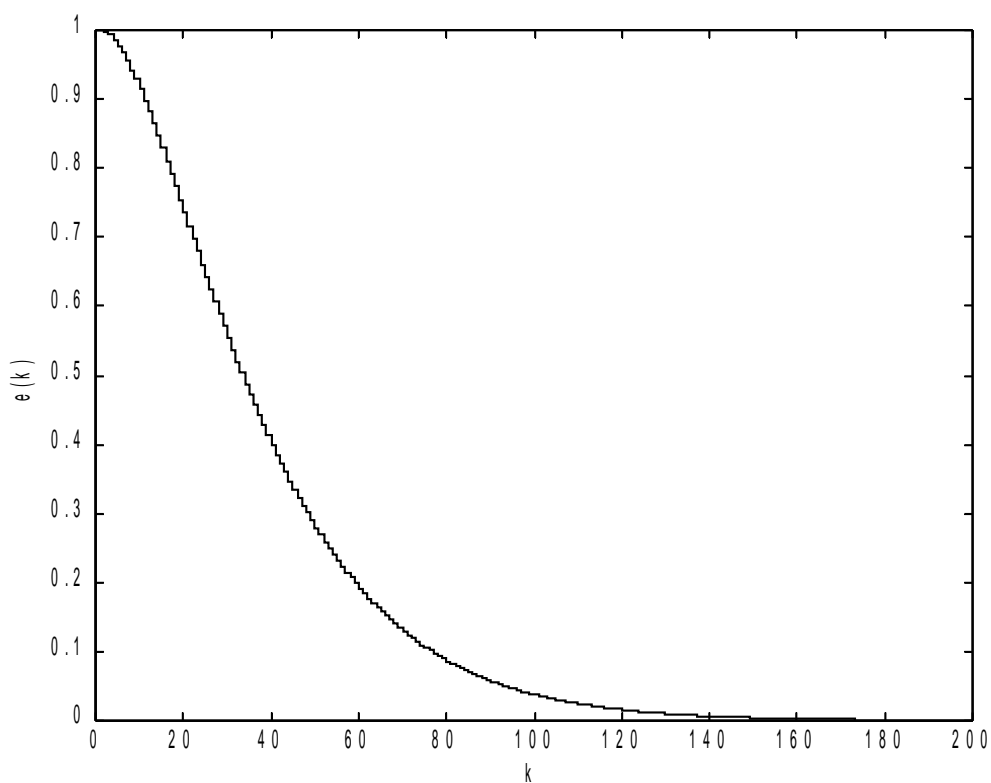
Po rozložení (3.8) na parciální zlomky získáme tvar

$$E(z) = \frac{e(0)z}{z + \frac{k_1}{2}} + \frac{\left(\frac{k_1}{2} + 1\right)e(0)z}{\left(z + \frac{k_1}{2}\right)^2} . \quad (3.9)$$

Po provedení zpětné Z-transformace na (3.9), získáme řešení diferenční rovnice (3.7) ve tvaru

$$e(k) = e(0) \left[\left(-\frac{k_1}{2}\right)^k + \left(\frac{k_1}{2} + 1\right)k \left(-\frac{k_1}{2}\right)^k \right] . \quad (3.10)$$

Průběh řešení (3.10) pro $e(0)=1$ a $k_1=-1,9$ je zobrazen na Obr. 3.4.



Obr. 3.4: Průběh řešení (3.10)

Funkci (3.10) je možné zapsat ve tvaru

$$e(k) = e(0) f(k) . \quad (3.11)$$

Funkce (3.11) je použita pro každou souřadnici pracovního prostoru. Pro každé dvě souřadnice i a j tedy platí

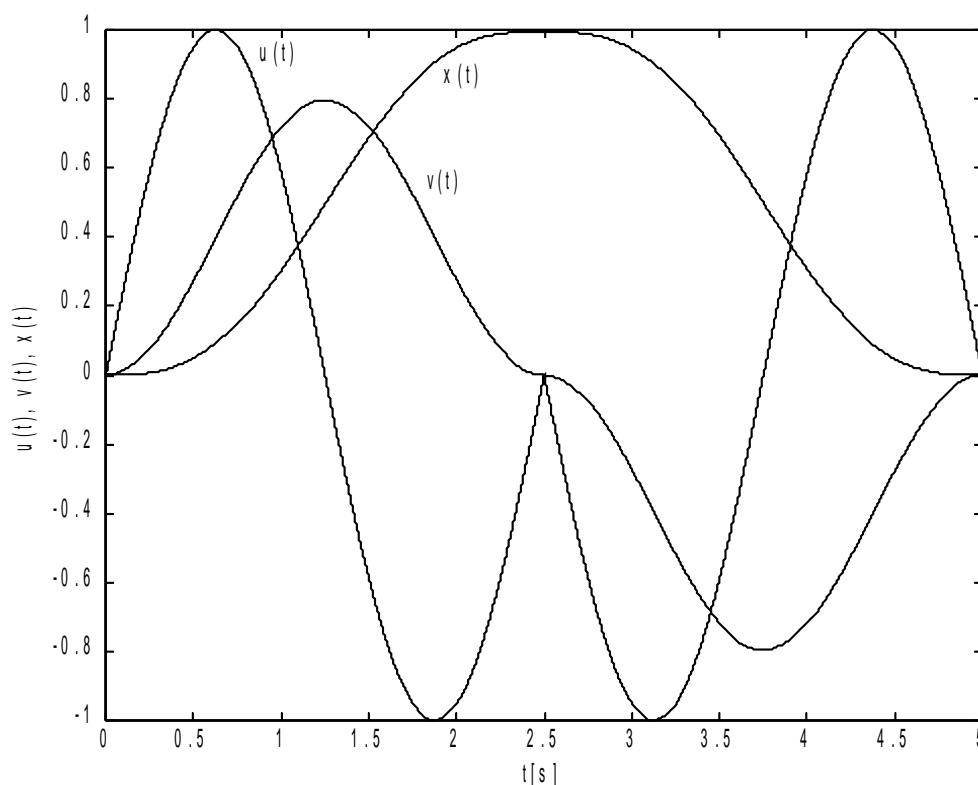
$$\frac{e_i(k)}{e_j(k)} = \frac{e_i(0)}{e_j(0)} = konst. \quad ,$$

z čehož je zřejmé, že trajektorie je přímkou.

3.4 Trénování online

V kapitole 3.2.2 je popsán základní postup získání trénovacích dat, pro učení sítě. Tento postup ovšem není vhodný pro skutečné mechanické systémy, které není možné umísťovat do předem zvolených pozic s předepsanými rychlostmi. Možným řešením by bylo umístit mechanismus do výchozí pozice a pomocí vhodného buzení ho rozpohybovat. Z dat získaných z tohoto pohybu by již bylo možné natrénovat síť a dále pak řídit mechanismus po zvolených trajektoriích a tak obdržet další data pro kvalitnější naučení sítě.

Nejprve je tedy nutné zvolit průběh prvotního buzení. Řízené mechanismy nejsou stabilní a proto by se mohly následkem špatného buzení lehce dostat do krajní pozice, což není žádoucí. Bylo by tedy vhodné vytvořit takové buzení, které by systémem vychýlilo z výchozí pozice a následně jej uvedlo zpět. Dalším požadavkem na buzení je jeho spojitost, jinak by docházelo k rázům. Pro buzení byla tedy zvolena harmonická funkce. Na Obr. 3.5 je znázorněn průběh buzení u , rychlosti v a polohy x pro lineární systém. Uvažované mechanismy jsou sice obecně nelineární, ale v malém okolí určitého bodu, je lze za lineární považovat. Toto buzení je postupně použito pro všechny členy mechanismu. Tímto postupem jsou získána data, potřebná k prvotnímu natrénování sítě.



Obr. 3.5: Průběhy buzení, rychlosti a polohy

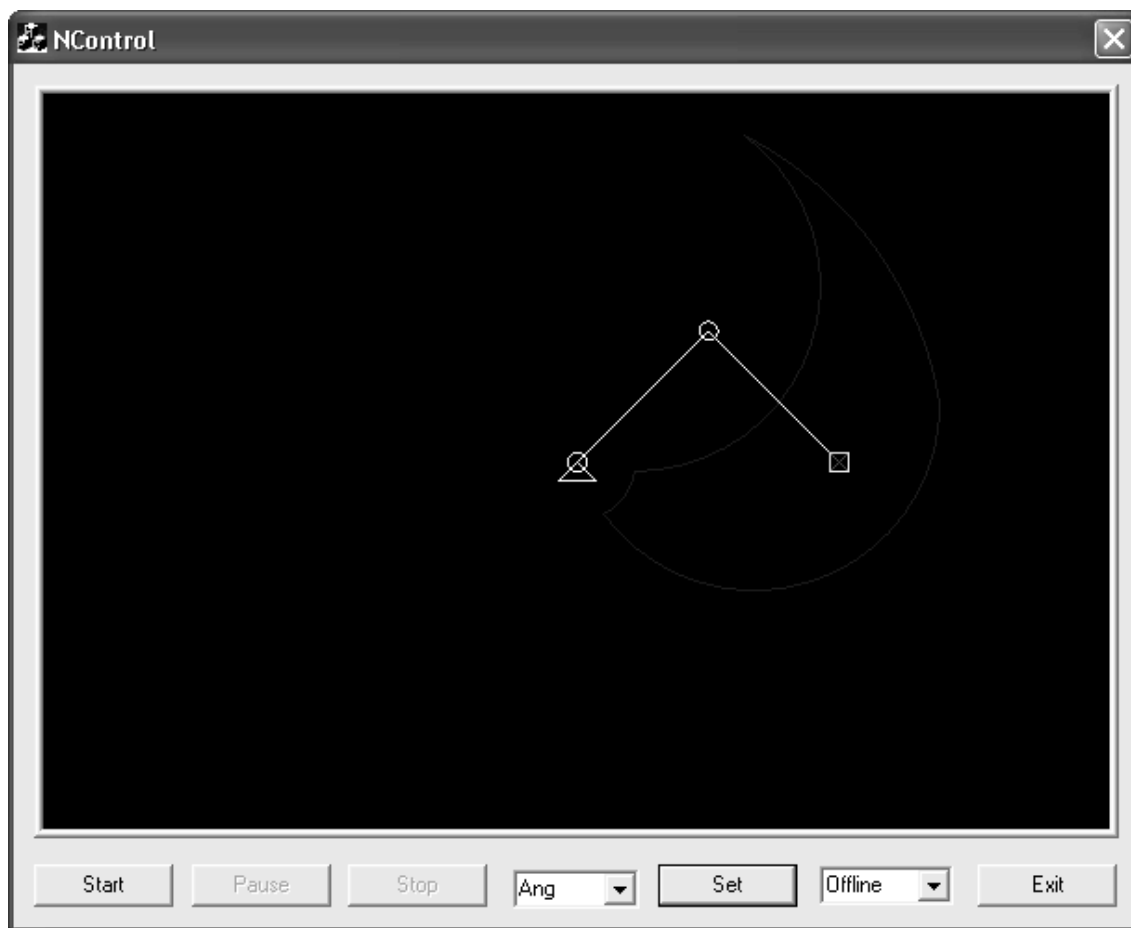
Pro prvotní trénování je použita síť bez skryté vrstvy. Důvodem je malé množství trénovacích dat a také potřeba mít natrénovanou síť v co nejkratším čase, protože v průběhu prvotního trénování je systém neřízen. Výhodou tedy je, pokud jsou řízené mechanismy ve výchozí pozici vyváženy tak, aby nedocházelo k samovolnému pohybu. Tato vlastnost je u vytvořených modelů splněna.

Po prvotním natrénování sítě, je již možné systém v okolí výchozí pozice řídit. Je tedy zvoleno kruhové okolí, ve kterém je náhodně vygenerováno několik bodů, do kterých mechanismus postupně najíždí. Tím se získají data pro trénování sítě, což umožní systém řídit ve větším okolí. Pro toto trénování je již použita síť s jednou skrytou vrstvou. Tento postup se opakuje, při zvětšování okolí, tak dlouho dokud není pokryt celý pracovní prostor mechanismu. Takto se, přirozeným postupem, získá natrénovaná síť, umožňující mechanismus řídit.

4 Praktická realizace

4.1 Aplikace NControl

V rámci práce byla vytvořena aplikace NControl, viz Obr. 4.1, ve které jsou implementovány výše uvedené simulační modely mechanismů a postupy řízení. Simulace pohybu modelů probíhá v reálném čase. Aplikace dále umožňuje ukládání průběhů řízení do textového souboru.



Obr. 4.1: Aplikace NControl

Pro tvorbu aplikace bylo zvoleno vývojové prostředí Microsoft Visual C++, a to zejména proto, že je kladen velký důraz na její výkon. Aplikace je tedy určena pro

operační systém Windows. Pro grafický výstup bylo použito rozhraní Direct3D, které umožňuje rychlé a kvalitní vykreslování modelů.

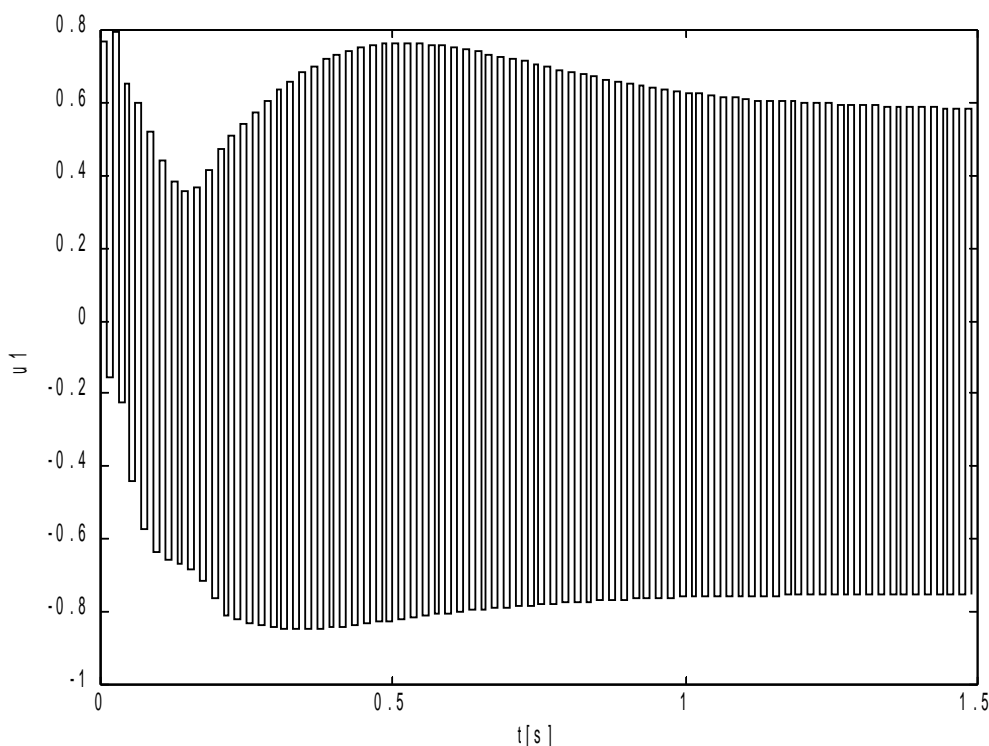
Aplikace byla provozována na počítači s následující konfigurací:

- Procesor – Intel Celeron M 370 (1,5 GHz, 400 MHz FSB, 1MB L2 cache)
- Paměť – 512MB DDR2
- Operační systém – Windows XP

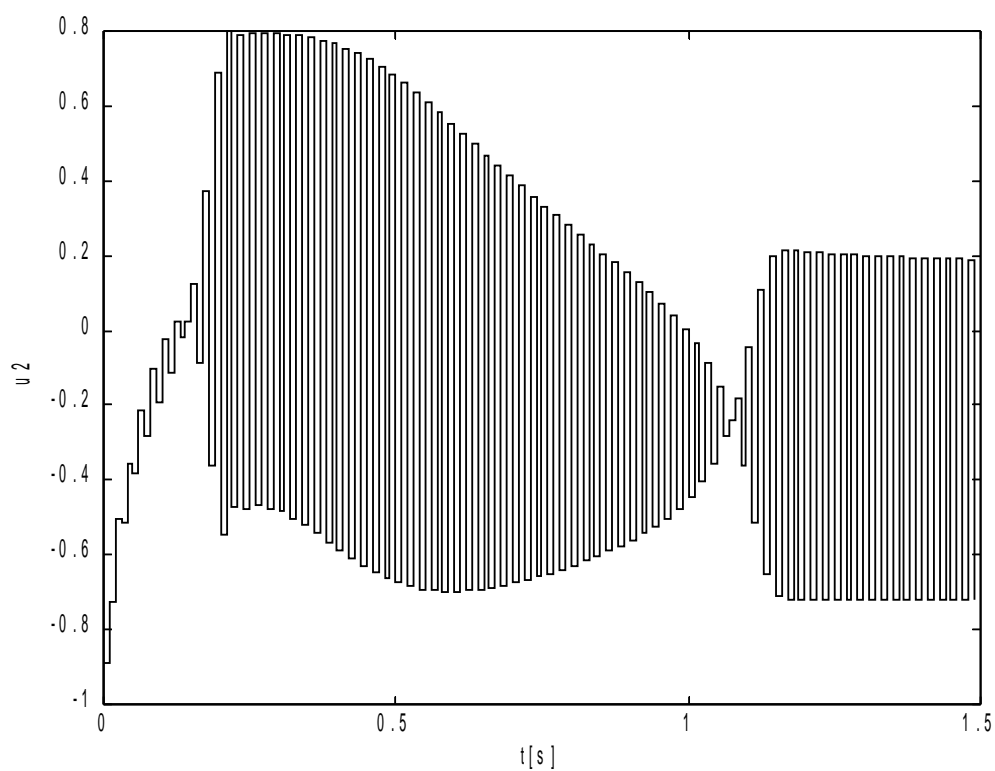
Této konfiguraci odpovídají časy trénování, které jsou v práci uváděny. Při provozu na počítači, s výrazně menším výkonem, by mohly nastat komplikace při trénování online, kde je mechanismus, při prvotním učení síť, po dobu trénování neřízen a mohl by se tedy dostat do krajní polohy. Vlastní řízení již nevyžaduje velký výkon počítače.

4.2 Dosažené průběhy řízení

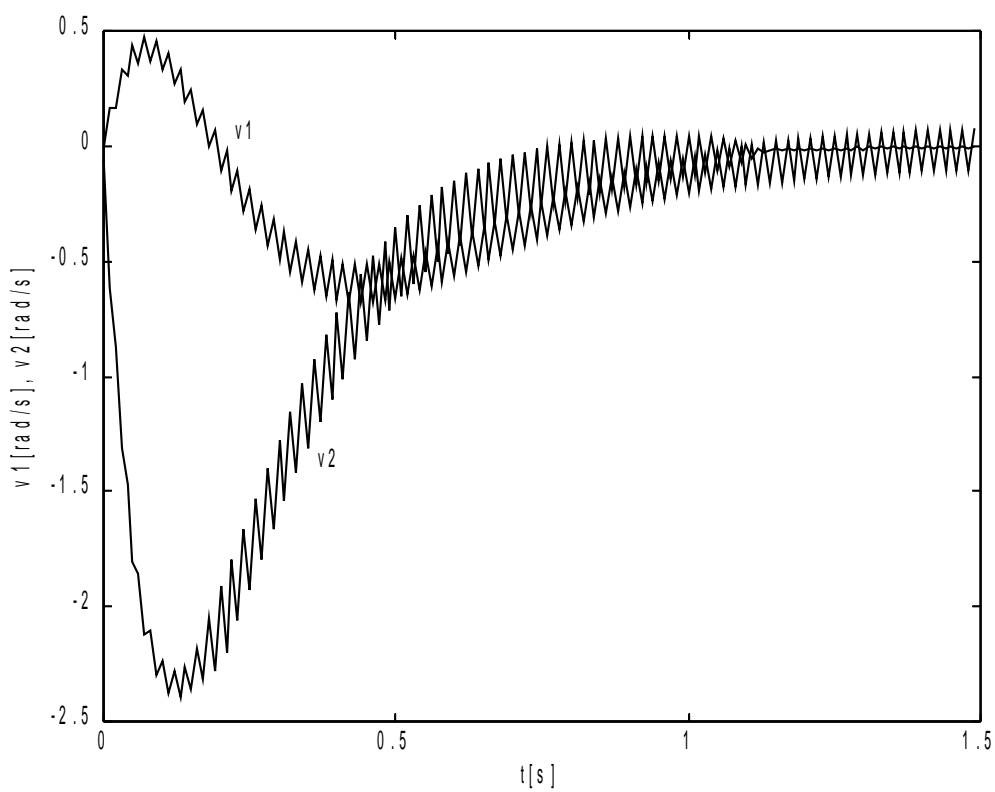
Na následujících grafech jsou postupně uvedeny průběhy řízení, rychlosti a pozice v zobecněných souřadnicích. Dále pak pozice a trajektorie v pracovním prostoru mechanismu. Průběhy byly získány při řízení angulárního robotu.



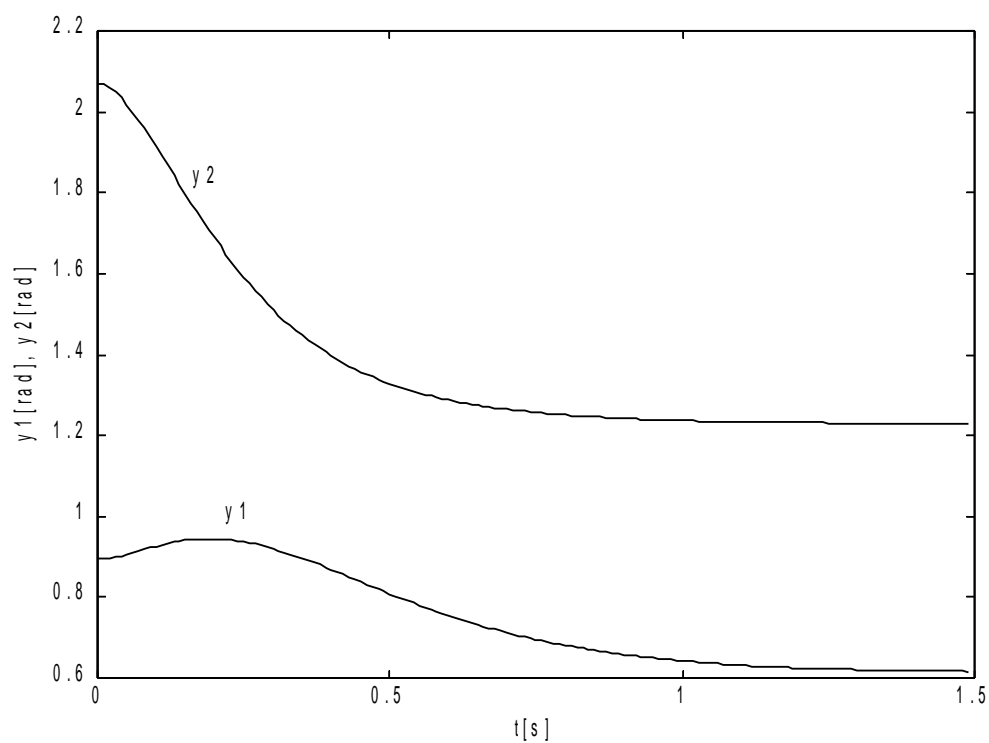
Obr. 4.2: Řízení u_1



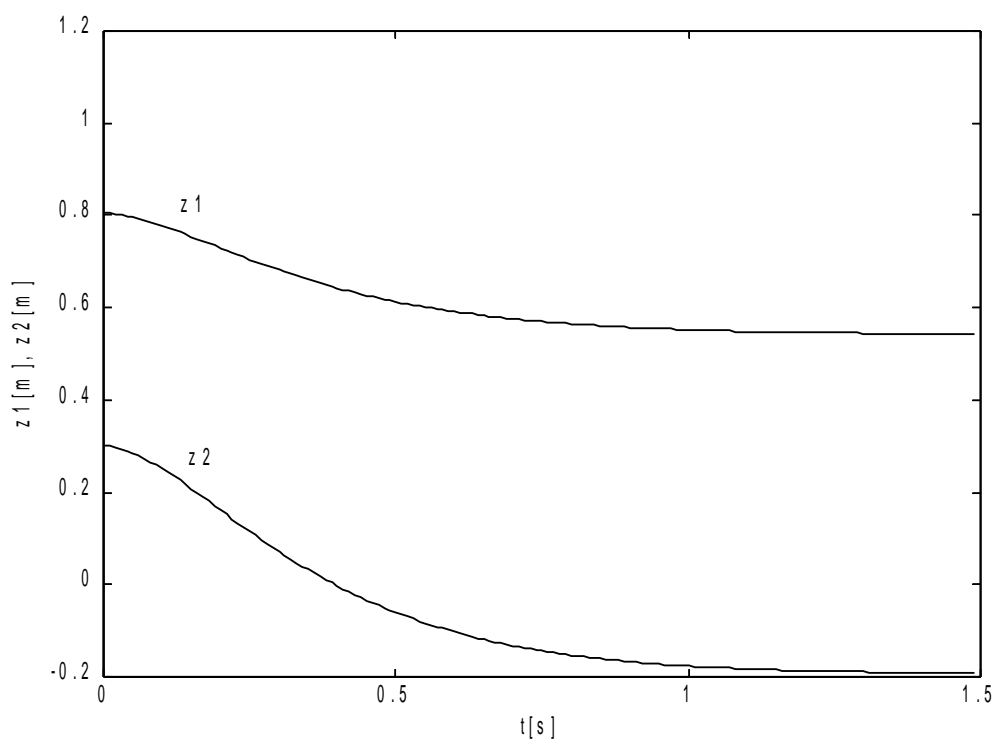
Obr. 4.3: Řízení u_2



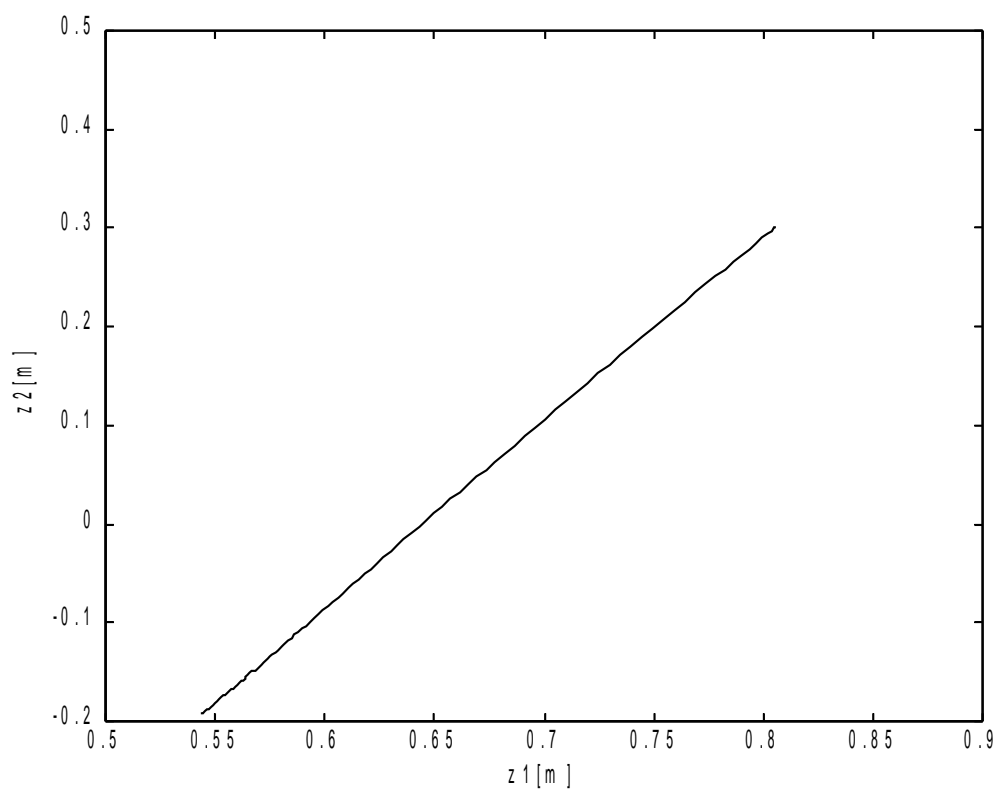
Obr. 4.4: Rychlosti v zobecněných souřadnicích



Obr. 4.5: Pozice v zobecněných souřadnicích



Obr. 4.6: Pozice v pracovním prostoru



Obr. 4.7: Trajektorie v pracovním prostoru

Závěr

V rámci práce byla nejprve implementována vícevrstvá perceptronová síť. Bylo vyzkoušeno několik optimalizačních metod pro minimalizaci chyby sítě.

Dále byly vytvořeny matematické modely čtyř rovinných mechanismů, umožňující simulaci jejich pohybu.

Byly navrženy postupy pro získání trénovacích dat a následné jejich využití pro řízení mechanismů. Tyto postupy byly úspěšně ověřeny na simulačních modelech. Vytvořené metody umožňují řízení mechanických systémů pouze na základě odpozorované závislosti dynamiky koncového bodu na řízení, jak bylo požadováno v zadání.

Veškeré popisované postupy jsou demonstrovány vytvořenou aplikací NControl, která umožňuje simulaci a řízení uvedených mechanismů, v reálném čase. Aplikace také umožňuje vypisování průběhů řízení do souboru.

Další možné rozšíření by mohlo spočívat v úpravě metod řízení tak, aby bylo možné měnit, u řízených mechanických systémů, hmotu v koncovém bodu, která představuje manipulované břemeno. Také by bylo vhodné ověřit funkci na prostorových mechanismech.

Seznam použité literatury

- [1] Mařík V., Štěpánková O., Lažanský J. a kol.: *Umělá inteligence (1)*, 1. vydání, 1993, ISBN 80-200-0496-3
- [2] Mařík V., Štěpánková O., Lažanský J. a kol.: *Umělá inteligence (4)*, 1. vydání, 2003, ISBN 80-200-1044-0
- [3] Press W. H., Teukolsky S. A., Vetterling W. T., Flannery B. P.: *Numerical Recipes in C*, 2. vydání, 1994, ISBN 0-521-43108-5
- [4] Hanuš B., Alexík M., Bíla J., Kroc L., Krupka Z.: *Teorie automatického řízení II.*, 1. vydání, 1985
- [5] Lubojacký O. a kol.: *Základy robotiky*, 2. vydání, 1990, ISBN 80-7083-034-4
- [6] Rektorys K. a spolupracovníci: *Přehled užití matematiky I*, 6. vydání, 1995, ISBN 80-85849-92-5
- [7] Vítečková M.: *Matematické metody v řízení L- a Z- transformace*, 1. vydání, 2002, ISBN 80-7078-570-5
- [8] Herout P.: *Učebnice jazyka C*, 3. vydání, 1999, ISBN 80-85828-21-9
- [9] Virius M.: *Od C k C++*, 1. vydání, 2000, ISBN 80-7232-110-2
- [10] Prosise J.: *Programování ve Windows pomocí MFC*, 1. vydání, 2002, ISBN 80-7226-309-9
- [11] Cvejn J.: *Efektivní způsob implementace operátorů v jazyku C++ pro třídy s dynamickou alokací dat*, K7 vědecko populární časopis Fakulty mechatroniky TU v Liberci, 2004, číslo 1, ISSN 1214-7370

PŘÍLOHY

Příloha A – Obsah CD

- Tato zpráva v elektronické podobě – **dokumentace.pdf**
- Aplikace NControl – **NControl.exe**
- Zdrojový kód aplikace NControl – **src**